

# Glossary

---

<b>Accelerated Graphics Port (AGP)</b>	A scalable architecture that increases the bandwidth available to a graphics controller and provides the performance necessary for a graphics controller to do texturing directly from system memory.
<b>Alpha Blending</b>	Uses a fourth color component which is not displayed but which corresponds to the opacity of a surface to control the amount of color of a pixel in the source surface to be blended with a pixel in the destination surface.
<b>Antialiasing</b>	An algorithm designed to reduce the stair-stepping artifacts (sometimes called jaggies) that result from drawing graphic primitives on a raster grid. The solution usually relies on the multi-bit raster's ability to display a number of pixel intensities. If the intensities of the neighboring pixels lie between the background and line intensities, the line becomes slightly blurred, and the jagged appearance is thereby diffused.
<b>API</b>	Application Programming Interface
<b>Bitmap</b>	A representation, consisting of rows and columns of dots, of a graphics image in computer memory. The value of each dot (whether it is filled in or not) is stored in one or more bits of data. For simple monochrome images, one bit is sufficient to represent each dot, but for colors and shades of gray, each dot requires more than one bit of data. The more bits used to represent a dot, the more colors and shades of gray that can be represented.
<b>BitPlane</b>	A rectangular array of bits mapped one-to-one with pixels. The framebuffer is a stack of bitplanes.
<b>Buffer</b>	A group of bitplanes that store a single component (such as depth or red) or a single index (such as the color index).
<b>Clipping</b>	A three dimensional operation that reduces the number of drawing calculations the CPU makes by eliminating any objects, or portions of objects, outside the viewing area.
<b>DDK</b>	Driver Development Kit
<b>Depth Cueing</b>	Reducing an object's color and intensity as a function of its distance from the observer. For instance, a bright, shiny red ball may look duller and darker the farther away it is from the observer.
<b>Direct3D (D3D)</b>	An Application Programming Interface (API) for manipulating and displaying 3-dimensional objects. Developed by Microsoft, Direct3D provides programmers with a way to develop 3-D programs abstracted from the hardware layer, but which can utilize 3-D capabilities of the underlying graphics accelerated hardware.

<b>DirectDraw</b>	Microsoft's new 2D library of graphics API's, enabling access to hardware's Blitting, clipping and flipping capabilities. DirectDraw provides low-level access to the frame buffer and advanced features of the display adapter.
<b>DirectDraw Video Port Extension (VPE)</b>	Microsoft's extension of DirectDraw to control the flow of data from a hardware video port device to a DirectDraw surface in video memory. As the VPE specification is finalized, it will be merged with the rest of the DirectDraw documentation.
<b>Direct Memory Execution (D.M.E.)</b>	Utilization of the entire AGP bandwidth through deep buffering and 2x side band signaling with write combining which provides the highest sustained data transfer rates across AGP.
<b>DLL</b>	Driver Link Library
<b>Double Buffering</b>	The process of using two frame buffers for smooth animation. Graphical contents of one frame buffer are displayed while updates occur on the other buffer. When the updates are complete, the buffers are switched. Only complete images are displayed, and the process of drawing is not shown. The result is the appearance of smooth animation.
<b>Fogging</b>	The alteration of the visibility of what is seen, depending on how far one is from the object.
<b>Frame Buffer</b>	A block of graphics memory that represents the display screen.
<b>GDI</b>	The Windows Graphics Device Interface, a library of video display and printer functions for 2D graphics.
<b>GFX</b>	Graphics Accelerator
<b>GFXGLDEV</b>	Device dependent part of the OpenGL ICD driver
<b>GFXGLICD</b>	Device independent portion of the OpenGL driver, which includes such items as the geometry and lighting engine.
<b>Gouraud shading</b>	Smooth interpolation of colors across a polygon or line segment. Colors are assigned at vertices and linearly interpolated across the primitive to produce a relatively smooth variation in color.
<b>H.324</b>	New communications standard for sharing video, voice and data over a single analog telephone line.
<b>HAL</b>	Hardware Abstraction Layer. A specification of a graphics hardware's functionality. Generally implemented into a device driver software program.
<b>Hyper Pipelined Architecture</b>	An architecture designed so that many operations are executed in parallel to improve performance.

<b>I<sup>2</sup>C*</b>	I <sup>2</sup> C* (Inter-Integrated Circuit) is a two-wire serial bus/protocol. A serial clock line (IICCLK) and serial data line (IICDAT) are used to transfer data between a bus master and a slave device. The maximum data rate is 100 Kbits/s. A slave may slow down the bus by inserting wait states. In the Intel740 graphics accelerator a single bus master can be implemented by using two of the Intel740 chip GPIO pins; one for IICCLK and one for IICDAT. Multiple slaves can be connected to this system (e.g., a TV tuner, video decoder, and digital TV encoder). However, only one I <sup>2</sup> C* master is allowed (Intel740 chip). The timing for the I <sup>2</sup> C* is derived from Intel740 chip PCI clock.
<b>ICD</b>	Installable Client Driver
<b>Lighting</b>	A mathematical formula for approximating the physical effect of light from various sources striking objects. Typical lighting models use light sources, an object's position & orientation and surface type.
<b>MCD</b>	Min-Client Driver, Microsoft's graphics interface which allows hardware acceleration of OpenGL at the rasterization level.
<b>Mipmapping</b>	When viewing a distant texture-mapped object in a 3D world, many texels make up each pixel seen on the screen, causing the textures to often appear aliased or distorted, if point sampling, the most common texture-mapping technique, is used. Mipmapping solves that problem by precomputing (that is, prefiltering) different levels of detail of your texture image, and accessing the appropriate level according to the object's distance from the camera. For example, a texture image which is 16x16 texels, will have four more mipmaps at lower resolutions, 8x8, 4x4, 2x2 and 1x1. Bilinear mipmapping chooses the closest mipmap image to your pixel's level of detail, then performs a bilinear interpolation upon that texture image to get the color value for the pixel. Trilinear mipmapping requires over twice the computational cost, as it chooses the two closest mipmaps, performs a bilinear interpolation on each, then averages the two results to arrive at the final screen pixel value.
<b>MMX™ technology</b>	A set of 57 multimedia instructions built into Intel's newest microprocessors. MMX™ technology-enabled microprocessors can handle many common multimedia operations, such as digital signal processing (DSP), that are normally handled by a separate sound or video card. However, only software especially written to call MMX™ technology instructions can take advantage of the MMX technology instruction set.
<b>OpenGL</b>	OpenGL, originally developed by Silicon Graphics Incorporated (SGI) for their graphics workstations; permits applications to create high-quality color images independent of windowing systems, operating systems, and hardware.
<b>OSR</b>	Operating Systems Release
<b>SDK</b>	Software Development Kit

<b>Pixel</b>	Short for picture element. The bits at location (x, y) of all the bitplanes in the framebuffer constitute the single pixel (x, y). It is the smallest discrete unit of a computer or TV tube that can be assigned a specific color, the “dots” that make up TV and computer screen pictures.
<b>POTS Video</b>	Low cost video conferencing over Plain Old Telephone Service (POTS).
<b>Raster</b>	A rectangular grid of picture elements, or pixels. The graphical data to be displayed on the raster is stored by the frame buffer. Raster operations can be performed on some portion or all of the raster. Such operations aid in the efficient handling of blocks of pixel data.
<b>Rendering</b>	The process of computing a graphical model's surface qualities, such as color, shading, smoothness, and texture, and creating a raster image.
<b>Setup</b>	Stage responsible for the precalculation of various derivatives used by inner loops of rendering algorithms.
<b>Shading</b>	The process of interpolating color within the interior of a polygon, or between the vertices of a line, during rasterization.
<b>Texel</b>	A texture element. A texel is obtained from texture memory and represents the color of the texture to be applied to a corresponding fragment.
<b>Texture antialiasing</b>	Bilinear or trilinear filtering. Also known as sub-texel positioning. If a pixel is between texels, the program chooses the color of the pixel by averaging the adjacent texels' colors instead of assigning it the exact color of one single texel. Without bilinear or trilinear filtering, the texture gets very blocky up close as multiple pixels get the exact same texel coloring, while the texture shimmers at a distance because small position changes keep producing large texel changes.
<b>Texture mapping</b>	The process of superimposing a 2-D texture or pattern over the surface of a 3-D graphical object. This is an efficient method of producing the appearance of texture, such as that of wood or stone, on a large surface area.
<b>Three Dimensional Graphics</b>	The display of objects and scenes with height, width, and depth information. The information is calculated in a coordinate system that represents three dimensions via x, y, and z axes.
<b>VxD</b>	Virtual Device Driver.
<b>WDM</b>	Win32 Driver Model (WDM) provides a common set of I/O services and binary-compatible device drivers for both Windows NT and future Windows operating systems. WDM will maximize system responsiveness and throughput by providing extremely low services and fewer ring transitions that interactive applications demand. All WDM drivers execute in Ring 0 and have access to low latency services. For backward compatibility, a Windows virtualization driver can be implemented to interface a hardware-specific legacy application to WDM.

**Z-buffer**

The depth buffer in 3-D graphics. The z-buffer memory locations, like those in the frame buffer, correspond to the pixels on the screen. The z-buffer, however, contains information relating only to the z-axis (or depth axis). The z-buffer is used in hidden surface removal algorithms, so that for each pixel written, the depth of the pixel is stored in the z-buffer. When subsequent objects attempt to draw that pixel, that object's z value is compared with the number in the z-buffer, and the write is omitted if the object is farther away from the eye.



# Index

---

## A

- AGP
  - Primer 2-55
  - Software Architecture 2-57
  - 2X AGP Support 2-3
  - 2X Interface 2-55
- AGP memory
  - 3D Pipeline 2-8
- Alpha Blending
  - Equation For 2-20
  - Functions 2-22
  - With DirectX 2-21
  - With OpenGL 2-21
- Alpha Testing 2-23
  - With DirectX 2-23
  - With OpenGL 2-23
- Antialiasing 2-40
  - With DirectX 2-41
  - With OpenGL 2-41

## B

- Back Face Culling 2-41
  - With DirectX 2-42
  - With OpenGL 2-42
- BitBLT
  - AGP 2-43
  - Color Expansion 2-44
- BitBLT Engine 2-42
- Buffers
  - Back buffer 2-38
  - Double Buffering 2-39
  - Front buffer 2-38
  - Pixel Formats and Buffers 2-38
  - Triple Buffering 2-39
  - Z-buffer 2-38
  - Z-buffering 2-39

## C

- Color Dithering 2-23
  - With DirectX 2-23
  - With OpenGL 2-23
- Color Keying
  - With DirectX 2-32

## D

- Device Driver Debugging Control 3-20
- Digital Camera
  - Video Conferencing 2-46
- Direct Memory Execution (DME) 2-2
- DirectX 1-1

- Color Key Capabilities 3-9
- Directdraw Hal Capabilities 3-8
- Direct3D Capabilities 3-12
- DIRECT3D RenderState Hardware / Software Support 3-16
- Driver Capabilities Of The Surface 3-11
- Driver Palette Capabilities 3-10
- Driver Software Architecture 3-5
- Driver-Specific Capabilities 3-9
- Driver-Specific Stretching and Effects Capabilities 3-10
- Functionality Control 3-20
- Mini Display Driver 3-6
- Supported RenderPrimitives 3-18
- Texture Capabilities 3-13
- DirectX Programming Environment 3-5
- Drawing Formats
  - Enabling with DirectX 2-38
  - Enabling with OpenGL 2-38
- DVD 1-1
- DVD Capabilities 2-50
  - Creating a VPE Port 2-51
  - Data Flow for DVD Playback 2-50
  - Data Flow Steps 2-50
  - Enabling Copy Protection Using SetMovieMode 2-53
  - MPEG-2 Movie Playback 2-50
  - Using TV Out with Copy Protection 2-52

## F

- FEATURES
  - Video In/Out Summary 1-2
  - 2D & Display 1-2
  - 3D Features Summary 1-2
- Features 1-2
- Fog
  - Density 2-18
  - Equation 2-17
  - OpenGL 2-18

## G

- GDI Escape Interface 3-20
- Gouraud Shading 2-24
  - Flat Shading vs. Gouraud Shading 2-24
  - With DirectX 2-24
  - With OpenGL 2-24

## H

- Hardware
  - parallelism 2-6
- Hardware Cursor 2-44

## I

Intercast 2-46

## O

OpenGL 1-1

Characteristics of Graphics Operations 3-3

ICD Buffer Allocation 3-2

ICD Driver Architecture 3-2

Independent Client Driver (ICD) 3-2

MCD Architecture 3-1

Mini Client Driver (MCD) 3-1

Programming Environment 3-1

Vertex Arrays 2-15

Vertex Information 2-15

OpenGL Drivers

Geometry Operations 3-3

OpenGL Feature Classification 4-27

OpenGL Programming

Antialiasing Application 4-33

Begin/End Paradigm 4-31

Bitmaps 4-32

Clipping 4-31

Colors and Coloring 4-32

Command Performance Ratings 4-34

Current Raster Position 4-31

Drawing, Reading, and Copying Pixels 4-33

Fog 4-33

Line Segments 4-32

Per-Fragment Operations 4-33

Pixel Rectangles 4-32

Points 4-32

Polygons 4-32

Rating OpenGL Features 4-29

Rectangles 4-31

Special Functions 4-34

Texturing 4-32

Vertex Arrays 4-31

Vertex Specification 4-31

Whole Framebuffer Operations 4-33

OpenGL Programming

Coordinate Transformation 4-31

OpenGL Programminig

OpenGL Feature Classification 4-27

## P

Parallel Data Processing (PDP) 2-3

Performance

Budgeting CPU Clock Cycles 4-18

Concurrency 4-2

CPU Cycle Targets 4-19

CPU Usage Model 4-2

Data Capture 4-20

Direct3D DrawPrimitive vs. Execute Buffers 4-11

D3D Performance vs. Buffer Size (Duty Cycle) 4-13

Formula 4-1

High Performance Transparency 4-17

Implications and Analysis 4-9

Improper Usage Model 4-2

Measurements 4-1

OpenGL Display Lists vs. Vertex Buffers 4-12

OpenGL Performance vs. Buffer Size (Duty Cycle) 4-14

Palette Changes 4-15

Procedural Texture Animation 4-16

Raster Speed Test Method 4-3

Result Summary 4-4

Screen Resolutions 4-18

Special Performance Considerations 4-11

Strategies 4-1

Tests Using Fog, Alpha Blending, Specular, and Anti-aliasing 4-7

Tests with Full Feature Sets 4-8

Tests With Gouraud Shading Levels 4-6

Texture Sizes 4-15

Triangle Packet Size 4-13

Video 4-19

Z Occlusion 4-10

Precise-Pixel Interpolation (PPI) 2-3

Primitive Types

Direct X 2-14

Primitives

DrawIndexPrimitive 2-14

OpenGL 2-15

Programming Tips

Avoiding Flipping Errors 4-27

Avoiding Stippling Errors 4-27

Dynamic AGP Buffer Placement 4-25

Minimizing State Transitions 4-24

OpenGL Programming 4-27

Optimal Artist Geometry Design 4-26

Optimal Artist Texture Design for Trilinear Filtering 4-26

Texture Sorting Is Not Required 4-27

Textures 4-21

Using Antialiasing 4-24

Using Color/chroma Keying on Top of Alpha Blended Textures 4-26

Using Mipmapping 4-25

Using Texture Palettes 4-25

Using Triple-Buffering 4-24

Using Z-Buffering 4-24

Z-Buffer Multi-pass Shadowing 4-23

## R

Related Documents 1-3

## S

Specular Highlighting

Specular Color 2-20

Specular highlighting 2-19

Stipple Pattern

With OpenGL 2-25

Stippled Pattern 2-25

With DirectX 2-25

Stretch BLT Engine 2-44



## T

- Texture Blending
  - DirectX Functions 2-30
  - OpenGL Modes & Equations 2-31
  - With OpenGL 2-31
- Texture Coordinate Mapping
  - U, V Space 2-11
- Textures
  - AGP Memory 2-25
  - Creating a Texture Surface with DirectX 2-27
  - Creating a Texture Surface with OpenGL 2-28
  - DirectX Texture Map Formats 2-27
  - DirectX Wrapping Formats 2-33
  - Filtering 2-34
  - Filtering With DirectX 2-35
  - Filtering With OpenGL 2-35
  - Mipmapping 2-36
  - Mipmapping with DirectX 2-37
  - Mipmapping with OpenGL 2-37
  - OpenGL Texture Map Formats 2-27
  - OpenGL Wrapping Formats 2-34
  - Point Filtering VS. Bilinear Filtering 2-35
- Transformations and Lighting 2-14
- TV Out
  - Sample Code Enabling Copy Protection Using VIDEOPARAMETERS 2-54
  - Sample Code for Enabling Copy Protection using VIDEOPARAMETERS 2-54
  - Sample Code for SetMovieMode 2-53
- TV Out Interface 2-51
  - Screen Modes Needed 2-51

## V

- VBI 1-1, 2-46
- VBI and Intercast 2-49
- Vertex Types
  - D3D 2-13
- Video 2-46

- Video Capture
  - VfW API 2-47
- Video Capture Port
  - VMI Interface 2-46
- Video Capture Programming 2-47
- Video Capture System Diagram 2-47
- Video Conferencing
  - Digital Camera 2-46
  - Plain Old Telephone Service (POTS) 2-46
- Video Display Modes 2-45
- Video Interface
  - VfW Capture Driver Capability 3-19
- VIDEO IN/OUT FEATURES
  - Summary 1-2
- Video Overlay 2-49
  - Field Based Content 2-49
  - Supported Data Formats 2-49

## W

- Win32 1-1

## Z

- Z-Buffering
  - With DirectX 2-40
  - With OpenGL 2-40
- 2D Capabilities 2-42
- 2D Sprites 2-31
- 3D Pipelien
  - Scan Converter 2-11
- 3D Pipeline
  - Color Calculator/Depth Test 2-11
  - Setup Engine 2-11
  - Texture Pipeline 2-11
- 3D pipeline
  - Implementation 2-6
- 3D pipeline unit 2-8

## **Intel around the world**

### **United States and Canada**

Intel Corporation  
Robert Noyce Building  
2200 Mission College Boulevard  
P.O. Box 58119  
Santa Clara, CA 95052-8119  
USA  
Phone: (800) 628-8686

### **Europe**

Intel Corporation (U.K.) Ltd.  
Pipers Way  
Swindon  
Wiltshire SN3 1RJ  
UK

Phone:

England	(44) 1793 403 000
Germany	(49) 89 99143 0
France	(33) 1 4571 7171
Italy	(39) 2 575 441
Israel	(972) 2 589 7111
Netherlands	(31) 10 286 6111
Sweden	(46) 8 705 5600

### **Asia Pacific**

Intel Semiconductor Ltd.  
32/F Two Pacific Place  
88 Queensway, Central  
Hong Kong, SAR  
Phone: (852) 2844 4555

### **Japan**

Intel Kabushiki Kaisha  
P.O. Box 115 Tsukuba-gakuen  
5-6 Tokodai, Tsukuba-shi  
Ibaraki-ken 305  
Japan  
Phone: (81) 298 47 8522

### **South America**

Intel Semicondutores do Brazil  
Rua Florida 1703-2 and CJ22  
CEP 04565-001 Sao Paulo-SP  
Brazil  
Phone: (55) 11 5505 2296

### **For More Information**

To learn more about Intel Corporation, visit our site  
on the World Wide Web at [www.intel.com](http://www.intel.com)

